INFORMATION AND COMPUTATION 76, 121–137 (1988)

The Equational Theory of Parameterized Specifications

PETER PADAWITZ

Fakultät für Informatik, Universität Passau, D-8390 Passau, West Germany

Equational theorems that are valid in a given parameterized data type can be derived in the inductive theory of the corresponding specification. Certain syntactical requirements to the specification guarantee that its equational theory agrees with the set of those equations which are satisfied by all "consistent" models of the specification. © 1988 Academic Press, Inc.

INTRODUCTION

Starting from a many-sorted signature SIG $= \langle S, \mathrm{OP} \rangle$ with sorts S and operation symbols OP an algebraic specification in the sense of Goguen *et al.* (1978) is given by a triple SPEC $= \langle S, \mathrm{OP}, E \rangle$, where E is a set of equations between OP-terms. The models of SPEC are all SPEC-algebras; i.e., all algebras with signature SIG which satisfy E. *Initial* SPEC-algebras play a dominant role: They realize the specification in a certain minimal way, usually expressed by the principles "no junk" and "no confusion." "No junk" means that each element a of an initial algebra A is "term-generated"; i.e., a is the interpretation in A of some variable-free term built up from operation symbols of SPEC. "No confusion" means that two such terms t and t' have the same interpretation in A *only if* the equation $t \equiv t'$ is derivable from the equations of SPEC. (The formal exposition can be found in Goguen *et al.*, 1978; Ehrig and Mahr, 1985; Goguen and Meseguer, 1985; Klaeren *et al.*, 1983).

From the model-theoretic point of view a SPEC-algebra A is initial if and only if A is term-generated and equations are valid in A only if they hold in *all* term-generated SPEC-algebras. Originally, initiality was formulated as a "universal" property in category theory: An object I in a category K (here, the category of all SPEC-algebras) is called *initial in* K if for every object A of K there is a unique morphism (here, homomorphism) from I to A (cf. Arbib and Manes, 1979).

Initial SPEC-algebras are isomorphic. Thus they have the same equational theory that consists of all equations, all ground term substitutions of which are derivable from E. Since the existence of such

0890-5401/88 $3.00

derivations is usually shown by induction, the equational theory of initial SPEC-algebras is called the *inductive theory* of SPEC. In general, inductive theories are not recursively enumerable and therefore the set of equations derivable by congruence and (fixed) induction rules is *properly* contained in the inductive theory of SPEC (cf. Nourani, 1981). The issue of this paper is to lift the connection between initiality and induction onto the level of parameterized specifications as they were studied by Thatcher *et al.*, 1982; Ehrig *et al.*, 1982; Ehrich, 1982; Lipeck, 1982; Ganzinger, 1983; Ehrig and Mahr, 1985; Padawitz, 1984, 1985, 1986.

A parameterized specification PAR is a pair of two specifications PSPEC and SPEC where the *parameter* PSPEC is part of the *target* SPEC. The role of initial algebras is taken over by a class of target algebras each of which is "freely generated" over a parameter algebra. These target algebras are called *PAR-algebras*. The corresponding class is denoted by Alg(PAR). Thus initiality is generated in a straightforward way, and inductive proof methods can be carried over from the non-parameterized to the parameterized case. Vice versa, PAR-algebras with empty parameter are initial SPEC-algebras.

If SPEC has sorts S, operation symbols OP, and equations E, we add an infinite set PX of parameter constants to OP and immediately obtain the following completeness result (Theorem 1.10): The equational theory of Alg(PAR) agrees with the inductive theory of $\text{SPEC(PX)} = \langle S, \text{OP} \cup \text{PX}, E \rangle$. In data type theory we are mostly concerned with a proper subclass of Alg(PAR) that evolves from parameter constraints which are not specifiable equationally. One of the most interesting constraints refers to the interpretation of a Boolean signature: PAR-algebras are restricted to those where the Boolean carrier is isomorphic to the two-valued model of propositional logic. In Section 2 we introduce "consistent" algebras which have two-element Boolean carriers and interpret (the characteristic functions of) equality predicates as identity relations. Under certain assumptions on the set E of equations the theory of consistent SPEC(PX)-algebras does not exceed the inductive theory of PAR. The assumptions are the following:

(1) E includes an axiomatization BOOL of Boolean algebra.

(2) (The characteristic functions of) equality predicates are specified as an S-sorted congruence relation.

(3) Operations with Boolean range and at least one Boolean argument are already in BOOL.

(4) Operations with non-Boolean range and at least one Boolean argument are "commutable" with other operations ("fork-compatibility").

The first two syntactical conditions make up the general assumption in

Section 1. Along with this requirement to E we add a rule to the equational calculus that allows us to replace an equation $\mathrm{eq}(t, t') \equiv \mathrm{true}$ by $t \equiv t'$ (cf. Definition 1.3). Conditions (3) and (4) are treated in Section 2.

In the preceding version of this paper (Padawitz, 1984) we started from consistent parameter algebras and presented criteria for maintaining consistency under parameterization. Although this concept provides a completeness theorem for the class of consistent PAR-algebras, it is limited to parameter-sorted equations and thus we omitted it here.

1. The Inductive Theory of Parameterized Specifications

A little familiarity with the syntax and semantics of equational specifications is assumed. So we recall only the basic notions and provide some additional notation.

A *signature* $\mathrm{SIG} = \langle S, \mathrm{OP} \rangle$ consists of a set S of *sorts* and a family $\mathrm{OP} = \{\mathrm{OP}_{ws} \mid w \in S^*, s \in S\}$ of sets of *operation symbols*. w and s are the *arity*, resp. *sort*, of $\sigma \in \mathrm{OP}_{ws}$. If w is empty, then σ is called a *constant*.

For S-sorted sets A and $s1, ..., sn \in S$, $A_{s1,...,sn}$ stands for $A_{s1} \times \cdots \times A_{sn}$. A *SIG-algebra* A consists of an S-sorted set—called the *carrier* of A and also denoted by A—and for all $w \in S^*$, $s \in S$, and $\sigma \in \mathrm{OP}_{ws}$, a function $\sigma^A: A_w \to A_s$ (or an element $\sigma^A \in A_s$ if w is empty). Let $X = \{X_s \mid s \in S\}$ be an infinite S-sorted set of variables. $T(SIG)$ denotes the algebra of *SIG-terms over X*. For all $t \in T(\mathrm{SIG})$, $op(t)$ and $var(t)$ stand for the sets of operation symbols, resp. variables, of t. $GT(SIG)$ denotes the algebra of ground or closed terms over SIG (terms without variables).

A *SIG-equation* $\langle l, r \rangle$ is a pair of SIG-terms with the same sort. We write $l \equiv r$ instead of $\langle l, r \rangle$. Let A be a SIG-algebra. A^X denotes the set of S-sorted functions $f = \{f_s: X_s \to A_s \mid s \in S\}$. Function application brackets are omitted when they are clear from the context. The homomorphic extension of f to $T(\mathrm{SIG})$ is also written f. Note that for all $t \in \mathrm{GT}(\mathrm{SIG})$ and $f, g \in A^X$ we have $ft = gt$ and thus write t^A instead of ft. A is *term-generated* if for all $a \in A^X$ some $t \in \mathrm{GT}(\mathrm{SIG})$ satisfies $t^A = a$.

A *satisfies* a SIG-equation $l \equiv r$, written $A \models l \equiv r$, if for all $f \in A^X$, $fl = fr$. This definition extends to classes of algebras and sets of equations as usual.

Let $\mathrm{SIG} = \langle S, \mathrm{OP} \rangle$ be a signature and E be a set of SIG-equations. Then SPEC is called a *specification*. $\equiv SPEC$ denotes the least congruence relation on $T(\mathrm{SIG})$ that contains E and is closed under the following substitution rule:

$$\text{For all } f \in T(\mathrm{SIG})^X, \qquad t \equiv \mathrm{SPEC}\ t' \text{ implies } ft \equiv \mathrm{SPEC}\ ft'.$$

The set of SIG-equations $l \equiv r$ with $fl \equiv \mathrm{SPEC}\, fr$ for all $f \in \mathrm{GT(SIG)}^X$ is called the *inductive theory of SPEC* and is denoted by *ITh(SPEC)*.

A SIG-algebra A is a *SPEC-algebra* ($A \in Alg(SPEC)$) if A satisfies E. *Gen(SPEC)* stands for the class of all term-generated SPEC-algebras.

The quotient algebras

$$I(SPEC) = \mathrm{GT(SIG)}/{\equiv}\mathrm{SPEC} \quad \text{and} \quad F(SPEC) = \mathrm{T(SIG)}/{\equiv}\mathrm{SPEC}$$

are called the *initial*, resp. *free*, *SPEC-algebras*. Here the restriction of $\equiv$SPEC to ground terms is also denoted by $\equiv$SPEC. (In category-theoretic terms, the pair $\langle \mathrm{F(SPEC)}, \eta: X \to \mathrm{F(SPEC)} \rangle$, where η assigns to $x \in X$ the $\equiv$SPEC-congruence class that contains x, is free over the S-sorted set X; i.e., for all SPEC-algebras A and $f \in A^X$ there is a unique SIG-homomorphism $f^*: \mathrm{F(SPEC)} \to A$ with $f^* \circ \eta = f \mid (\mathrm{SPEC})$ is an initial object in Alg(SPEC); i.e., $\langle \mathrm{I(SPEC)}, \eta: \varnothing \to \mathrm{I(SPEC)} \rangle$ is free over the empty set.)

The following theorem is folklore in data type theory (cf. Ehrig and Mahr, 1985; Goguen and Meseguer, 1985).

THEOREM 1.1. *Let* $\mathrm{SPEC} = \langle S, \mathrm{OP}, E \rangle$ *be a specification,* $\mathrm{SIG} = \langle S, \mathrm{OP} \rangle$ *and* $l \equiv r$ *be a* SIG-*equation.*

(1) $\mathrm{Alg(SPEC)} \models l \equiv r$ *iff* $\mathrm{F(SPEC)} \models l \equiv r$ *iff* $l \equiv \mathrm{SPEC}\ r$.

(2) $\mathrm{Gen(SPEC)} \models l \equiv r$ *iff* $\mathrm{I(SPEC)} \models l \equiv r$ *iff* $(l \equiv r) \in \mathrm{ITh(SPEC)}$.

(3) *For all* $A \in \mathrm{Alg(SPEC)}$ *there is a unique* SIG-*homomorphism from* I(SPEC) *to* A.

A *parameterized specification* is a pair of two specifications PSPEC and SPEC such that PSPEC is componentwise included in SPEC.

EXAMPLE 1.2. Let BOOL be a specification of Boolean algebras; i.e., BOOL consists of a sort bool, constants true and false, operation symbols for logical connectives, and a complete set of Boolean algebra axioms. We give a parameterized specification of $\langle \mathrm{DATA}, \mathrm{SET} \rangle$ of finite sets over DATA-algebras. $\sigma \in \mathrm{OP}_{ws}$ is now written as $\sigma: w \to s$. x, y, z, b, s, s', s'' are variables. The parameter specification DATA is

DATA = BOOL +
  sorts: entry
  opns: eq: entry, entry → bool
  eqns: $\mathrm{eq}(x, x) \equiv \mathrm{true}$
    $\mathrm{eq}(x, y) \equiv \mathrm{eq}(y, x)$
    $(\mathrm{eq}(x, y) \wedge \mathrm{eq}(y, z)) \Rightarrow \mathrm{eq}(x, z) \equiv \mathrm{true}$

and the target specification reads as

SET = DATA +
sorts: set
opns: $\varnothing$: $\to$ set
ins: set, entry $\to$ set
if: bool, set, set $\to$ set
has: set, entry $\to$ bool
del: set, entry $\to$ set
eqs: set, set $\to$ bool
eqns:

$$\mathrm{ins}(\mathrm{ins}(s, x), x) \equiv \mathrm{ins}(s, x) \tag{e1}$$
$$\mathrm{ins}(\mathrm{ins}(s, x), y) \equiv \mathrm{ins}(\mathrm{ins}(s, y), x) \tag{e2}$$
$$\mathrm{ins}(\mathrm{if}(b, s, s'), x) \equiv \mathrm{if}(b, \mathrm{ins}(s, x), \mathrm{ins}(s', x)) \tag{e3}$$
$$\mathrm{if}(\mathrm{true}, s, s') \equiv s \tag{e4}$$
$$\mathrm{if}(\mathrm{false}, s, s') \equiv s' \tag{e5}$$
$$\mathrm{has}(\varnothing, x) \equiv \mathrm{false} \tag{e6}$$
$$\mathrm{has}(\mathrm{ins}(s, x), y) \equiv \mathrm{eq}(x, y) \vee \mathrm{has}(s, y) \tag{e7}$$
$$\mathrm{has}(\mathrm{if}(b, s, s'), x) \equiv (b \Rightarrow \mathrm{has}(s, x)) \wedge (\neg b \Rightarrow \mathrm{has}(s', x)) \tag{e8}$$
$$\mathrm{del}(\varnothing, x) \equiv \varnothing \tag{e9}$$
$$\mathrm{del}(\mathrm{ins}(s, x), y) \equiv \mathrm{if}(\mathrm{eq}(x, y), \mathrm{del}(s, y), \mathrm{ins}(\mathrm{del}(s, y), x)) \tag{e10}$$
$$\mathrm{del}(\mathrm{if}(b, s, s'), x) \equiv \mathrm{if}(b, \mathrm{del}(s, x), \mathrm{del}(s', x)) \tag{e11}$$
$$\mathrm{eqs}(s, s) \equiv \mathrm{true} \tag{e12}$$
$$\mathrm{eqs}(s, s') \equiv \mathrm{eqs}(s', s) \tag{e13}$$
$$(\mathrm{eqs}(s, s') \wedge \mathrm{eqs}(s', s'')) \Rightarrow \mathrm{eqs}(s, s'') \equiv \mathrm{true} \tag{e14}$$
$$(\mathrm{eqs}(s, s') \wedge \mathrm{eq}(x, y)) \Rightarrow \mathrm{eqs}(\mathrm{ins}(s, x), \mathrm{ins}(s', y)) \equiv \mathrm{true} \tag{e15}$$
$$(\mathrm{eqs}(s, s') \wedge \mathrm{eq}(x, y) \wedge \mathrm{has}(s, x)) \Rightarrow \mathrm{has}(s', y) \equiv \mathrm{true} \tag{e16}$$
$$(\mathrm{eqs}(s, s') \wedge \mathrm{eq}(x, y)) \Rightarrow \mathrm{eqs}(\mathrm{del}(s, x), \mathrm{del}(s', y)) \equiv \mathrm{true} \tag{e17}$$
$$\mathrm{eqs}(\mathrm{if}(b, s, s'), s'') \equiv (b \Rightarrow \mathrm{eqs}(s, s'')) \wedge (\neg b \Rightarrow \mathrm{eqs}(s', s'')) \tag{e18}$$
$$\mathrm{eqs}(\mathrm{ins}(s, x), s') \equiv \mathrm{has}(s', x) \wedge \mathrm{eqs}(\mathrm{del}(s, x), \mathrm{del}(s', x)) \tag{e19}$$

"ins" inserts an element into a set, "has" asks for the containment of an element in a set; "del" (delete) removes an element from a set. In initial semantics, (e4) and (e5) would specify "if" completely. But here the initial SET-algebra I(SET) is useless because $\mathrm{I(SET)}_{\mathrm{entry}}$ is empty. We are not interested in I(SET) but in the class of $\langle$DATA, SET$\rangle$-algebras (see below). Equations (e3), (e8), (e11), and (e18) guarantee that SET-algebras are "fork-compatible" (cf. Definition 2.2 below).

A further particularity of SET is the involvement of equality predicates eq and eqs for the sorts entry, resp. set, and corresponding congruence axioms, e.g., (e12)–(e17). This is our

General Assumption. Let SPEC = $\langle S, \mathrm{OP}, E\rangle$ be a specification. BOOL (cf. Example 1.2) is a subspecification of SPEC and for all

$s \in S$-{bool}, $\mathrm{OP}_{s,s,\mathrm{bool}}$ contains an operation symbol eq_s called the *equality predicate for s*. Moreover, for all $s1, ..., sn, s \in S$-{bool}, $\sigma \in \mathrm{OP}_{s1,...,sn,s}$, $\tau \in \mathrm{OP}_{s1,...,sn,\mathrm{bool}}$, and some variables $x, y, z, x_1, ..., x_n, y_1, ..., y_n$ the following equations called *equality axioms* are in E:

$$\mathrm{eq}_s(x, x) \equiv \mathrm{true}$$
$$\mathrm{eq}_s(x, y) \equiv \mathrm{eq}_s(y, x)$$
$$(\mathrm{eq}_s(x, y) \wedge \mathrm{eq}_s(y, z)) \Rightarrow \mathrm{eq}_s(x, z) \equiv \mathrm{true}$$
$$(\mathrm{eq}_{s1}(x_1, y_1) \wedge \cdots \wedge \mathrm{eq}_{sn}(x_n, y_n)) \Rightarrow \mathrm{eq}_s(\sigma(x_1, ..., x_n), \sigma(y_n, ..., y_n)) \equiv \mathrm{true}$$
$$(\mathrm{eq}_{s1}(x_1, y_1) \wedge \cdots \wedge \mathrm{eq}_{sn}(x_n, y_n) \wedge \tau(x_1, ..., x_n)) \Rightarrow \tau(y_1, ..., y_n) \equiv \mathrm{true}.$$

Equality predicates affect semantics and proof theory as follows:

DEFINITION 1.3. Let $\mathrm{SPEC} = \langle S, \mathrm{OP}, E \rangle$ be a specification and $\mathrm{SIG} = \langle S, \mathrm{OP} \rangle$. A SIG-algebra A is *equality-compatible* if for all $s \in S$-{bool} and $a, b \in A_s$,

$$a = b \qquad \text{if } \mathrm{eq}_s^A(a, b) = \mathrm{true}^A.$$

≈SPEC stands for the least congruence relation on T(SIG) that contains E and is closed under the following rules:

(1) For all $f \in T(\mathrm{SIG})^X$, $t \approx \mathrm{SPEC}\ t'$ implies $ft \approx \mathrm{SPEC}\ ft'$.

(2) For all $s \in S$-{bool}, $\mathrm{eq}_s(t, t') \approx \mathrm{SPEC}$ true implies $t \approx \mathrm{SPEC}_s\ t'$.

EAlg(SPEC) denotes the class of equality-compatible SPEC-algebras.
EGen(SPEC) stands for the class of equality-compatible and term-generated SPEC-algebras.

$$EF(SPEC) = \mathrm{T(SIG)}/\approx \mathrm{SPEC}, \text{ resp. } EI(SPEC) = \mathrm{GT(SIG)}/\approx \mathrm{SPEC},$$

denote the *free*, resp. *initial equality-compatible*, *SPEC-algebra*. (Here the restriction of ≈SPEC to ground terms is also denoted by ≈SPEC.) The *equality-compatible inductive theory*, *EITh(SPEC)*, is the set of SIG-equations $t \equiv t'$ such that for all $f \in \mathrm{GT(SIG)}^X$, $ft \approx \mathrm{SPEC}\ ft'$.

In other words, a SIG-algebra A is equality-compatible iff for all $s \in S$-{bool}, A satisfies the *conditional* equation

$$\mathrm{eq}_s(x, y) \equiv \mathrm{true} \Rightarrow x \equiv y.$$

Hence Theorem 1.1 is carried over to equality-compatible algebras and their congruence relation ≈SPEC:

THEOREM 1.4. *Let* $\mathrm{SPEC} = \langle S, \mathrm{OP}, E \rangle$ *be a specification*, $\mathrm{SIG} = \langle S, \mathrm{OP} \rangle$, *and* $l \equiv r$ *be a* SIG-*equation.*

(1) $\mathrm{EAlg(SPEC)} \models l \equiv r$ *iff* $\mathrm{EF(SPEC)} \models l \equiv r$ *iff* $l \approx \mathrm{SPEC}\ r$.

(2) $\mathrm{EGen(SPEC)} \models l \equiv r$ *iff* $\mathrm{EI(SPEC)} \models l \equiv r$ *iff* $(l \equiv r) \in \mathrm{EITh(SPEC)}$.

(3) *For all* $A \in \mathrm{EAlg(SPEC)}$ *there is a unique* SIG-*homomorphism from* EI(SPEC) *to* A.

Let $\mathrm{PAR} = \langle \mathrm{PSPEC}, \mathrm{SPEC} \rangle$ be a parameterized specification such that SPEC satisfies our general assumption. We denote the *forgetful functor* from EAlg(SPEC) to EAlg(PSPEC) by EU_{PAR} and the *free functor* from EAlg(PSPEC) to EAlg(SPEC) by EF_{PAR}. (For "non-categorists," if K and K' are two categories, a *functor* $U\colon K \to K'$ maps objects of K to objects of K' and morphisms of K to morphisms of K' such that for all morphisms $f\colon A \to B$ and $g\colon B \to C$ in K, $U(g \circ f) = U(g) \circ U(f)$. If for each object A in K' there is a free object $\langle F^A, \eta^A\colon A \to U(B) \rangle$ with $F^A \in K$ (see above), then the assignment of F^A to A can be extended to a functor $F\colon K' \to K$ called the *free functor*. U is the corresponding *forgetful functor*. For details see Arbib and Manes (1979).

DEFINITION 1.5. Let $\mathrm{PAR} = \langle \mathrm{PSPEC}, \mathrm{SPEC} \rangle$ be a parameterized specification and $A \in \mathrm{EAlg(PSPEC)}$. Each equality-compatible SPEC-algebra that is isomorphic to $\mathrm{EF}_{\mathrm{PAR}}(A)$ is called a *PAR-algebra*. The *data type specified by PAR* is the class of all PAR-algebras denoted by *Alg(PAR)*.

EXAMPLE 1.6 (cf. Example 1.2). Let A be a DATA-algebra such that A_{bool} consists of two distinct elements true^A and false^A, and eq^A is the identity on A_{entry}. Define an equality-compatible SET-algebra B as follows:

$$
\begin{aligned}
&B_s = A_s \quad \text{for } s \in \{\mathrm{bool}, \mathrm{entry}\},\\
&B_{\mathrm{set}} = \{M \subseteq A_{\mathrm{entry}} \mid M \text{ finite}\},\\
&\sigma^B = \sigma^A \quad \text{for } \sigma = \mathrm{eq} \text{ and all operation symbols in BOOL},\\
&\varnothing^B = \varnothing,\\
&\mathrm{ins}^B(M, a) = M \cup \{a\},\\
&\mathrm{has}^B(M, a) = (a \in M),\\
&\mathrm{del}^B(M, a) = M - \{a\},\\
&\mathrm{eqs}^B(M, M') = (M = M'),\\
&\mathrm{if}^B(\mathrm{true}^A, M, M') = M,\\
&\mathrm{if}^B(\mathrm{false}^A, M, M') = M'.
\end{aligned}
$$

We claim that B is a $\langle \mathrm{DATA}, \mathrm{SET} \rangle$-algebra, more precisely: B and $\mathrm{EF}_{\langle \mathrm{DATA}, \mathrm{SET} \rangle}(A)$ are isomorphic. It is sufficient to show that $\langle B, \mathrm{id}^A \rangle$ is a free object over A with respect to $EU_{\langle \mathrm{DATA}, \mathrm{SET} \rangle}$; i.e., for all $C \in \mathrm{EAlg(SET)}$ and DATA-homomorphisms $h\colon A \to EU_{\langle \mathrm{DATA}, \mathrm{SET} \rangle}(C)$ there is a unique

SET-homomorphism $h^*\colon B \to C$ with $\mathrm{EU}_{\langle \mathrm{DATA,SET} \rangle}(h^*) = h$. The proof is straightforward (cf. Thatcher *et al.*, Sec. 5, 1982).

DEFINITION 1.7. Let PSPEC $= \langle \mathrm{PS}, \mathrm{POP}, \mathrm{PE} \rangle$, SPEC $= \langle S, \mathrm{OP}, E \rangle$, PAR $= \langle \mathrm{PSPEC}, \mathrm{SPEC} \rangle$ be a parameterized specification and PC be an additional PS-sorted set of constants such that there is a bijection from PC to the set PX of PS-sorted variables of X. Then the *inductive theory of PAR, ITh(PAR)*, is given by the restriction of the equality-compatible inductive theory of

$$SPEC(PC) = \langle S, \mathrm{OP} \cup \mathrm{PC}, E \rangle$$

to SIG-terms.

We now investigate the relationship between ITh(PAR) and equations satisfied by all PAR-algebras. Is ITh(PAR) complete with respect to this class? The following "initial representation" of PAR-algebras is useful to answer this question.

THEOREM 1.8. *Let* PAR $= \langle \mathrm{PSPEC}, \mathrm{SPEC} \rangle$ *be a parameterized specification and A be an equality-compatible* PSPEC-*algebra such that for all parameter sorts s,* OP_s *and A_s are disjoint.* Let $SIG(A) = \langle S, \mathrm{OP} \cup A \rangle$. The *diagram of A, $\Delta(A)$, consists of all equations $\sigma(a) \equiv \sigma^A(a)$ with $\sigma \in \mathrm{POP}$ and $a \in A_{\mathrm{arity}(\sigma)}$. Note that both sides of such an equation are terms over the signature* SIG(A). *On the lefthand side, "a" is used as a list of constants, while on the righthand side, "a" is a value and the value $\sigma^A(a)$ of σ^A at a is again treated as a constant. Hence the diagram of A represents the interpretation of* POP *in A as a—possibly infinite—set of equations. Now let*

$$SPEC(A) = \langle S, \mathrm{OP} \cup A, E \cup \Delta(A) \rangle$$

and $B = EF_{\mathrm{PAR}}(A)$. Clearly, B becomes a SIG(A)-*algebra by defining $a^B = \eta^A(a)$ for all $a \in A$ where η^A is the "unit morphism" of the free object $\langle B, \eta^A\colon A \to \mathrm{EU}_{\mathrm{PAR}}(B) \rangle$ over A. Moreover, B is an equality-compatible initial* SPEC(A)-*algebra; i.e.,* $\mathrm{EF}_{\mathrm{PAR}}(A)$ *and* EI(SPEC(A)) *are isomorphic as* SPEC-*algebras.*

Proof. First we show that B satisfies $\Delta(A)$. Let $\sigma \in \mathrm{POP}$, $a \in A_{\mathrm{arity}(\sigma)}$, and $f \in B^X$. Then

$$\begin{aligned} f(\sigma(a)) &= \sigma^B(f(a)) = \sigma^B(a^B) = \sigma^B(\eta^A(a)) \\ &= \eta^A(\sigma^A(a)) = (\sigma^A(a))^B = f(\sigma^A(a)). \end{aligned}$$

Hence B satisfies $\sigma(a) \equiv \sigma^A(a)$. Now let C be an equality-compatible SPEC(A)-algebra. We need a unique SIG(A)-homomorphism $h\colon B \to C$.

Define $g: A \to \mathrm{EU}_{\mathrm{PAR}}(C)$ by $g(a) = a^C$ for all $a \in A$. Since $\langle B, \eta^A \rangle$ is a free object over A, g uniquely extends to a SIG-homomorphism $g^*: B \to C$ such that $\mathrm{EU}_{\mathrm{PAR}}(g^*) \circ \eta^A = g$. Hence

$$g^*(a^B) = g^*(\eta^A(a^A)) = g(a^A) = a^C$$

for all $a \in A$. Thus g^* is a SIG(A)-homomorphism. Since every SIG(A)-homorphism from B to C extends g, g^* is the only one. ∎

LEMMA 1.9. *Let* PSPEC = ⟨PS, POP, PE⟩, PAR = ⟨PSPEC, SPEC⟩ *be a parameterized specification and* PC *be a* PS-*sorted set of constants as in Definition* 1.7. $\mathrm{EF}_{\mathrm{PAR}}$(EF(PSPEC)) *and the initial equality-compatible* SPEC(C)-*algebra* EI(SPEC(PC)) *are isomorphic as* SPEC-*algebras.*

Proof. Let PAR0 = ⟨⟨PS, ∅, ∅⟩, PSPEC⟩, PAR1 = ⟨⟨PS, ∅, ∅⟩, SPEC⟩, $A = \mathrm{EF}_{\mathrm{PAR0}}(\mathrm{PC})$, and $B = \mathrm{EF}_{\mathrm{PAR1}}(\mathrm{PC})$. It is well-known that A is isomorphic to the free equality-compatible PSPEC-algebra EF(PSPEC), while Theorem 1.8 implies that B and EI(SPEC(PC)) are isomorphic. Since $\mathrm{EF}_{\mathrm{PAR1}}$ and $\mathrm{EF}_{\mathrm{PAR}} \circ \mathrm{EF}_{\mathrm{PAR0}}$ are naturally isomorphic (cf. Mac Lane, Sec. IV.8, 1972), we conclude that $\mathrm{EF}_{\mathrm{PAR}}$(EF(PSPEC)) and B are isomorphic. This gives the statement of the corollary. ∎

From 1.8 and 1.9 we derive the following completeness theorem for the class of all PAR-algebras.

THEOREM 1.10. *Let* PAR = ⟨PSPEC, SPEC⟩ *be a parameterized specification and* SIG *be the signature of* SPEC. *For all* SIG-*equations* $l \equiv r$,

$$\mathrm{Alg(PAR)} \models l \equiv r \qquad \textit{iff} \quad (l \equiv r) \in \mathrm{ITh(PAR)}.$$

Proof. By Lemma 1.9, EI(SPEC(PC)) ∈ Alg(PAR). Hence Alg(PAR) ⊨ $l \equiv r$ implies $(l \equiv r) \in$ ITh(PAR). Vice versa, suppose that $l \equiv r$ is in the inductive theory of PAR, and let $B \in$ Alg(PAR). Then by Theorem 1.8, there is an equality-compatible PSPEC-algebra A such that EI(SPEC(A)) and B are isomorphic. To show that EI(SPEC(A)) and thus B satisfy $l \equiv r$, let $f \in \mathrm{GT(SIG}(A))^X$. There are $g \in \mathrm{GT(SIG(PC))}^X$ and h: PC → A such that $f = h \circ g$. Since $(l \equiv r) \in$ ITh(PAR), we have $gl \approx \mathrm{SPEC(PC)}\, gr$. Thus

$$fl = hgl \approx \mathrm{SPEC}(A) hgr = fr.$$

Hence B satisfies $l \equiv r$. ∎

EXAMPLE 1.11 (cf. Example 1.2). The equation

$$\mathrm{has(ins}(s, x), x) \equiv \mathrm{true} \qquad \text{(e20)}$$

can be derived from (e7) and $\mathrm{eq}(x, x) \equiv \mathrm{true}$ and therefore belongs to the inductive theory of $\langle \mathrm{DATA}, \mathrm{SET} \rangle$. Thus by Theorem 1.10, all $\langle \mathrm{DATA}, \mathrm{SET} \rangle$-algebras satisfy (e20).

In order to prove that an equation is in the inductive theory of PAR it is sufficient to induce on terms without parameter-sorted operation symbols:

Proposition 1.12. $(l \equiv r) \in \mathrm{ITh}(\mathrm{PAR})$ *iff all* $f \in \mathrm{T}(\mathrm{SIG})^X$ *with* $\mathrm{var}(f(X)) \subseteq \mathrm{PX}$, $f(\mathrm{PX}) = \mathrm{PX}$ *and*, $\mathrm{sort}(\mathrm{op}(f(X - \mathrm{PX}))) \subseteq S - \mathrm{PS}$ satisfy $fl \approx \mathrm{SPEC}\, fr$.

Proof. Let $g \in \mathrm{GT}(\mathrm{SIG}(\mathrm{PC}))^X$. Then there are $f \in T(\mathrm{SIG})^X$ and $h \in \mathrm{GT}(\mathrm{SIG}(\mathrm{PC}))^X$ such that $h \circ f = g$ and f is as above. Hence $fl \approx \mathrm{SPEC}\, fr$ implies $gl \approx \mathrm{SPEC}(\mathrm{PC})\, gr$. ∎

Example 1.13 (cf. Example 1.2). To show that the equation

$$\mathrm{has}(\mathrm{del}(s, x), y) \equiv \neg\mathrm{eq}(x, y) \wedge \mathrm{has}(s, y) \tag{e21}$$

is in the inductive theory of $\langle \mathrm{DATA}, \mathrm{SET} \rangle$ we apply Proposition 1.12: It is sufficient to prove

$$\mathrm{has}(\mathrm{del}(t, x), y) \approx \mathrm{SPEC}(\mathrm{PX})\ \neg\mathrm{eq}(x, y) \wedge \mathrm{has}(t, y)$$

for all $t \in \mathrm{GT}(\mathrm{SIG}(\mathrm{PX}))$ with $\mathrm{op}(t) \subseteq \{\varnothing, \mathrm{ins}, \mathrm{if}, \mathrm{del}\}$. We conclude from Eqs. (e9)–(e11) of SET that for all $t \in \mathrm{GT}(\mathrm{SIG}(\mathrm{PX}))$ with $\mathrm{del} \in \mathrm{op}(t)$ there is t' with $t \approx \mathrm{SPEC}(\mathrm{PX}) t'$ and $\mathrm{del} \notin \mathrm{op}(t')$. Thus the following three cases are left where $\equiv$ means $\approx \mathrm{SPEC}(\mathrm{PX})$.

Case 1. $t = \varnothing$. Then

$$\begin{aligned} \mathrm{has}(\mathrm{del}(t, x), y) &\equiv \mathrm{has}(\varnothing, y) \equiv \mathrm{false} \equiv \neg\mathrm{eq}(x, y) \wedge \mathrm{false} \\ &\equiv \neg\mathrm{eq}(x, y) \wedge \mathrm{has}(t, y). \end{aligned}$$

Case 2. $t = \mathrm{ins}(u, z)$. Then

$$\begin{aligned} &\mathrm{has}(\mathrm{del}(t, x), y) \\ &\quad \equiv \mathrm{has}(\mathrm{if}(\mathrm{eq}(z, x), \mathrm{del}(u, x), \mathrm{ins}(\mathrm{del}(u, x), z)), y) \\ &\quad \equiv (\neg\mathrm{eq}(z, x) \vee \mathrm{has}(\mathrm{del}(u, x), y)) \\ &\qquad \wedge (\mathrm{eq}(z, x) \vee \mathrm{has}(\mathrm{ins}(\mathrm{del}(u, x), z), y)) \\ &\quad \equiv (\neg\mathrm{eq}(z, x) \vee \mathrm{has}(\mathrm{del}(u, x), y)) \\ &\qquad \wedge (\mathrm{eq}(z, x) \vee \mathrm{eq}(z, y) \vee \mathrm{has}(\mathrm{del}(u, x), y)) \\ &\quad \equiv (\neg\mathrm{eq}(z, x) \wedge (\mathrm{eq}(z, x) \vee \mathrm{eq}(z, y))) \vee \mathrm{has}(\mathrm{del}(u, x), y) \\ &\quad \equiv (\neg\mathrm{eq}(z, x) \wedge \mathrm{eq}(z, y)) \vee \mathrm{has}(\mathrm{del}(u, x), y) \\ &\quad \equiv (\neg\mathrm{eq}(z, x) \wedge \mathrm{eq}(z, y)) \vee (\neg\mathrm{eq}(x, y) \wedge \mathrm{has}(u, y)) \\ &\qquad\qquad \text{(by induction hypothesis)} \end{aligned}$$

$$
\begin{aligned}
&\equiv (\neg \mathrm{eq}(x, y) \wedge \mathrm{eq}(z, y)) \vee (\neg \mathrm{eq}(x, y) \wedge \mathrm{has}(u, y)) \\
&\equiv \neg \mathrm{eq}(x, y) \wedge (\mathrm{eq}(z, y) \vee \mathrm{has}(u, y)) \\
&\equiv \neg \mathrm{eq}(x, y) \wedge \mathrm{has}(t, y).
\end{aligned}
$$

Case 3. $t = \mathrm{if}(z, u, v)$. Then

$$
\begin{aligned}
&\mathrm{has}(\mathrm{del}(t, x), y) \\
&\quad \equiv \mathrm{has}(\mathrm{if}(z, \mathrm{del}(u, x), \mathrm{del}(v, x)), y) \\
&\quad \equiv (\neg z \vee \mathrm{has}(\mathrm{del}(u, x), y)) \wedge (z \vee \mathrm{has}(\mathrm{del}(v, x), y)) \\
&\quad \equiv (\neg z \vee (\neg \mathrm{eq}(x, y) \wedge \mathrm{has}(u, y))) \wedge (z \vee (\neg \mathrm{eq}(x, y) \wedge \mathrm{has}(v, y))) \\
&\qquad\quad \text{(by induction hypothesis)} \\
&\quad \equiv \neg \mathrm{eq}(x, y) \wedge (\neg z \vee \mathrm{has}(u, y)) \wedge (z \vee \mathrm{has}(v, y)) \\
&\quad \equiv \neg \mathrm{eq}(x, y) \wedge \mathrm{has}(t, y).
\end{aligned}
$$

By Theorem 1.10, all $\langle$DATA, SET$\rangle$-algebras satisfy (e21).

2. Consistent Algebras

We are now concerned with the following subclasses of Alg(SPEC), resp. Gen(SPEC):

Definition 2.1. Let SPEC be a specification with signature SIG. An equality-compatible SIG-algebra is called *consistent* if A_{bool} consists of two distinct elements true^A and false^A. The class of consistent, resp. consistent and term-generated, SPEC-algebras is denoted by *CAlg(SPEC)*, resp. *CGen(SPEC)*.

In the sequel we develop the analogon of Theorem 1.4 (1) and (2) for consistent algebras. Completeness of $\approx$SPEC with respect to CAlg(SPEC) depends on the specification of if–then–else operators, more generally, of "fork operators," which have Boolean arguments, but a non-Boolean range:

Definition 2.2. Let $s1, ..., sn$, $s \in S$, and $\sigma \in \mathrm{OP}_{s1, ..., sn, s}$ such that for some $1 \leqslant i \leqslant n$, $si = \mathrm{bool}$. If $s = \mathrm{bool}$, then σ is *logical.* If $s \neq \mathrm{bool}$, then σ is called a *fork operation.* SPEC is *fork-compatible* if for all fork operations τ and terms

$$t = \sigma(x_1, ..., x_j, \tau(y_1, ..., y_k), x_{j+1}, ..., x_n),$$

where σ is neither fork nor logical, there is $t' \in T(\mathrm{SIG})$ such that $t \approx \mathrm{SPEC}t'$, $\mathrm{var}(t') \subseteq \{x_1, ..., x_n, y_1, ..., y_k\}$ and for each subterm $\sigma' u$ of t' with σ' neither fork nor logical, u is a list of variables.

In SET (cf. Example 1.2) all logical operations are in BOOL and "if" is the only fork operation. Fork-compatibility of SET is guaranteed by equations (e3), (e8), (e11), and (e18).

The following lemma provides congruence relations on SPEC-algebras to transform these into consistent algebras.

LEMMA 2.3. *Suppose that* $\mathrm{SPEC} = \langle S, \mathrm{OP}, E \rangle$ *is a fork-compatible specification with signature* SIG, A *being an equality-compatible* SPEC-*algebra and* $l \equiv r$ *being a* SIG-*equation such that all logical operations of* SIG *are in* BOOL *and* A *does not satisfy* $l \equiv r$. *Then there is* SIG-*congruence relation* $\sim$ *on* A *such that* $A/\sim$ *is a consistent* SPEC-*algebra and does not satisfy* $l \equiv r$.

Proof. By assumption, there is $f \in A^X$ with $fl \neq fr$. Hence $\mathrm{eq}_s^A(fl, fr) \neq \mathrm{true}^A$ because A is equality-compatible. Since A_{bool} is a Boolean algebra, some prime ideal I of A_{bool} contains $\mathrm{eq}_s^A(fl, fr)$ (cf. Bell and Slomson, 3.4, 1971; Rasiowa and Sikorski, 1.8 and 11.5, 1968; Richter, 1.5.11, 1978). An S-sorted equivalence relation $\sim$ on A is inductively defined as follows:

(1) $\sim_{\mathrm{bool}} = \{\langle a, b \rangle \in A_{\mathrm{bool}} \mid a, b \in I \text{ or } a, b \notin I\}$,

(2) for all $s \in S$-$\{\mathrm{bool}\}$ and $a, b \in A_s$, $\mathrm{eq}_s^A(a, b) \sim \mathrm{true}^A$ implies $a \sim b$,

(3) for all $w \in S^+$, $s \in S$, fork operations $\sigma \in \mathrm{OP}_{ws}$, and $a, b \in A_w$, $a \sim b$ implies $\sigma^A(a) \sim \sigma^A(b)$.

The congruence property of $\sim$ is checked by induction on its definition: Let $s1, ..., sn, s \in S$, $\sigma \in \mathrm{OP}_{s1,...,sn,s}$, and for all $1 \leqslant i \leqslant n$, $a_i, b_i \in A_{si}$ such that $a_i \sim b_i$. We must show

(4) $\sigma^A(a_1, ..., a_n) \sim \sigma^A(b_1, ..., b_n)$.

Case 1. σ is fork. Then (4) follows from (3).

Case 2. σ is logical. By assumption, σ is in BOOL. Hence (4) follows from (1).

Case 3. For all $1 \leqslant i \leqslant n$, $si \neq \mathrm{bool}$.

Case 3.1. For all $1 \leqslant i \leqslant n$, $\mathrm{eq}_s^A(a_i, b_i) \sim \mathrm{true}^A$. Then (4) follows from (2) and the equality axioms.

Case 3.2. There are a fork operation τ, $1 \leqslant j \leqslant n$, and $c_1, ..., c_k, d_1, ..., d_k \in A$ such that $a_j = \tau^A(c_1, ..., c_k)$, $b_j = \tau^A(d_1, ..., d_k)$, and for all $1 \leqslant i \leqslant m$, $c_i \sim d_i$. Since SPEC is fork-compatible, there is $t \in T(\mathrm{SIG})$ such that

$$\sigma(x_1, ..., x_{j-1}, \tau(y_1, ..., y_k), x_{j+1}, ..., x_n) \approx \mathrm{SPEC}t,$$

$\mathrm{var}(t) \subseteq \{x_1, ..., x_{j-1}, x_{j+1}, ..., x_n, y_1, ..., y_k\}$, and for each subterm $\sigma' u$ of t with σ' neither fork nor logical, u is a list of variables. Let $f, g \in A^X$ satisfy

$$f(x_i) = a_i \quad \text{and} \quad g(x_i) = b_i \quad \text{for all } 1 \leqslant i < j \text{ and } j < i \leqslant n$$

and

$$f(y_i) = c_i \quad \text{and} \quad g(y_i) = d_i \quad \text{for all } 1 \leqslant i \leqslant k.$$

By induction hypothesis, for each subterm $\sigma' u$ of t with σ' neither fork nor logical, $\sigma'^A(fu) \sim \sigma'^A(gu)$. Hence Cases 1 and 2 imply

$$\sigma^A(a_1, ..., a_n) = ft \sim gt = \sigma^A(b_1, ..., b_n).$$

Thus $\sim$ is a SIG-congruence relation on A. Let $B = A/\sim$. From (1) we conclude that B_{bool} consists of two distinct elements true^B and false^B. By (2), B is equality-compatible. Hence B is consistent. B is a SPEC-algebra because A satisfies E and $\sim$ is SIG-congruent. Finally, $\mathrm{eq}_s^A(fl, fr) \in I$ implies $\mathrm{eq}_s^A(fl, fr) \not\sim \mathrm{true}^A$ and thus $fl \not\sim fr$ because $fl \sim fr$ would imply

$$\mathrm{eq}_s^A(fl, fr) \sim \mathrm{eq}_s^A(fr, fr) = \mathrm{true}^A.$$

Therefore B does not satisfy $l \equiv r$. ∎

Fork-compatibility implies that the classes of equality-compatible and consistent SPEC-algebras have the same equational theory:

THEOREM 2.4. *Let* SPEC $= \langle S, \mathrm{OP}, E \rangle$ *be a fork-compatible specification with signature* SIG *such that all logical operations of* SIG *are in* BOOL. *For all* SIG-*equations* $l \equiv r$,

(1) CAlg(SPEC) $\models l \equiv r$ *iff* EF(SPEC) $\models l \equiv r$ *iff* $l \approx \mathrm{SPEC}\, r$.

(2) CGen(SPEC) $\models l \equiv r$ *iff* EI(SPEC) $\models l \equiv r$ *iff* $(l \equiv r) \in$ EITh(SPEC).

Proof. (1) Let $l \approx \mathrm{SPEC}\, r$. Since CAlg(SPEC) is a subclass of EAlg(SPEC), CAlg(SPEC) $\models l \equiv r$ follows from Theorem 1.4 (1). Vice versa, suppose that $l \approx \mathrm{SPEC}\, r$ does not hold. Then by Theorem 1.4 (1), EF(SPEC) does not satisfy $l \equiv r$. Therefore Lemma 2.3 provides some consistent quotient of EF(SPEC) which does not satisfy $l \equiv r$.

(2) is proved like (1): Replace CAlg(SPEC), EF(SPEC), and $\approx$SPEC by CGen(SPEC), EI(SPEC), and EITh(SPEC), respectively, and note that quotients of EI(SPEC) are term-generated. ∎

From Theorems 1.10 and 2.4 (2) we infer that Alg(PAR) and CGen(SPEC(PC)) have the same equational theory provided that SPEC is fork-compatible:

THEOREM 2.5. *Let* PSPEC = ⟨PS, POP, PE⟩, SPEC = ⟨S, OP, E⟩, PAR = ⟨PSPEC, SPEC⟩ *be a parameterized specification and* PC *be an infinite* PS*-sorted set of additional constants such that* SPEC *is fork-compatible and all logical operations of* SIG = ⟨S, OP⟩ *are in* BOOL. *Then for all* SIG*-equations* $l \equiv r$, $\mathrm{Alg(PAR)} \models l \equiv r$ *iff* $(l \equiv r) \in \mathrm{ITh(PAR)}$ *iff* $\mathrm{CGen(SPEC(PC))} \models l \equiv r$. ▮

EXAMPLE 2.6 (cf. Example 2.2). To show that the equation

$$\mathrm{del}(\mathrm{del}(s, x), y) \equiv \mathrm{del}(\mathrm{del}(s, y), x) \qquad \text{(e22)}$$

is in the inductive theory of ⟨DATA, SET⟩ we apply Proposition 1.12: It is sufficient to prove

$$\mathrm{del}(\mathrm{del}(t, x), y) \approx \mathrm{SPEC(PX)}\ \mathrm{del}(\mathrm{del}(t, y), x)$$

for all $t \in \mathrm{GT(SIG(PX))}$ with $\mathrm{op}(t) \subseteq \{\varnothing, \mathrm{ins}, \mathrm{if}, \mathrm{del}\}$. We conclude from equations (e9)–(e11) of SET that for all $t \in \mathrm{GT(SIG(PX))}$ with $\mathrm{del} \in \mathrm{op}(t)$ there is t' with $t \approx \mathrm{SPEC(PX)}t'$ and $\mathrm{del} \notin \mathrm{op}(t')$. Thus the following three cases are left where $\equiv$ means $\approx \mathrm{SPEC(PX)}$.

Case 1. $t = \varnothing$. Then

$$\mathrm{del}(\mathrm{del}(t, x), y) \equiv \mathrm{del}(\varnothing, y) \equiv \varnothing \equiv \mathrm{del}(\varnothing, x) \equiv \mathrm{del}(\mathrm{del}(t, y), x).$$

Case 2. $t = \mathrm{ins}(u, z)$. Then

$$\begin{aligned}
&\mathrm{del}(\mathrm{del}(t, x), y)\\
&\quad \equiv \mathrm{del}(\mathrm{if}(\mathrm{eq}(z, x), \mathrm{del}(u, x), \mathrm{ins}(\mathrm{del}(u, x), z)), y)\\
&\quad \equiv \mathrm{if}(\mathrm{eq}(z, x), \mathrm{del}(\mathrm{del}(u, x), y), \mathrm{del}(\mathrm{ins}(\mathrm{del}(u, x), z), y))\\
&\quad \equiv \mathrm{if}(\mathrm{eq}(z, x), \mathrm{del}(\mathrm{del}(u, x), y),\\
&\qquad\qquad \mathrm{if}(\mathrm{eq}(z, y), \mathrm{del}(\mathrm{del}(u, x), y), \mathrm{ins}(\mathrm{del}(\mathrm{del}(u, x), y), z)))\\
&\quad \equiv \mathrm{if}(\mathrm{eq}(z, x), \mathrm{del}(\mathrm{del}(u, y), x),\\
&\qquad\qquad \mathrm{if}(\mathrm{eq}(z, y), \mathrm{del}(\mathrm{del}(u, y), x), \mathrm{ins}(\mathrm{del}(\mathrm{del}(u, y), x), z)))\\
&\qquad\qquad \text{(by induction hypothesis)}\\
&\quad \equiv \mathrm{if}(\mathrm{eq}(z, x), t1, \mathrm{if}(\mathrm{eq}(z, y), t1, t2)) = t3, \qquad (1)
\end{aligned}$$

where $t1 = \mathrm{del}(\mathrm{del}(u, y), x)$ and $t2 = \mathrm{ins}(\mathrm{del}(\mathrm{del}(u, y), x), z)$. On the other hand,

$$\begin{aligned}
&\mathrm{del}(\mathrm{del}(t, y), x)\\
&\quad \equiv \mathrm{del}(\mathrm{if}(\mathrm{eq}(z, y), \mathrm{del}(u, y), \mathrm{ins}(\mathrm{del}(u, y), z)), x)\\
&\quad \equiv \mathrm{if}(\mathrm{eq}(z, y), \mathrm{del}(\mathrm{del}(u, y), x), \mathrm{del}(\mathrm{ins}(\mathrm{del}(u, y), z), x))\\
&\quad \equiv \mathrm{if}(\mathrm{eq}(z, y), \mathrm{del}(\mathrm{del}(u, y), x),\\
&\qquad\qquad \mathrm{if}(\mathrm{eq}(z, x), \mathrm{del}(\mathrm{del}(u, y), x), \mathrm{ins}(\mathrm{del}(\mathrm{del}(u, y), x), z)))\\
&\quad \equiv \mathrm{if}(\mathrm{eq}(z, y), t1, \mathrm{if}(\mathrm{eq}(z, x), t1, t2)) = t4. \qquad (2)
\end{aligned}$$

Furthermore,

$$
\begin{aligned}
&\mathrm{eqs}(t3, t4)\\
&\equiv (\mathrm{eq}(z, x) \Rightarrow \mathrm{eqs}(t1, \mathrm{if}(\mathrm{eq}(z, y)\, t1, \mathrm{if}(\mathrm{eq}(z, x), t1, t2))))\\
&\quad \wedge (\neg \mathrm{eq}(z, x) \Rightarrow \mathrm{eqs}(\mathrm{if}(\mathrm{eq}(z, y), t1, t2), \mathrm{if}(\mathrm{eq}(z, y), t1, \mathrm{if}(\mathrm{eq}(z, x), t1, t2))))\\
&\equiv (\mathrm{eq}(z, x) \Rightarrow ((\mathrm{eq}(z, y) \Rightarrow \mathrm{eqs}(t1, t1)\\
&\qquad\qquad \wedge (\neg \mathrm{eq}(z, y) \Rightarrow \mathrm{eqs}(t1, \mathrm{if}(\mathrm{eq}(z, x), t1, t2))))\\
&\quad \wedge (\neg \mathrm{eq}(z, x) \Rightarrow ((\mathrm{eq}(z, y) \Rightarrow \mathrm{eqs}(t1, \mathrm{if}(\mathrm{eq}(z, y), t1, \mathrm{if}(\mathrm{eq}(z, x), t1, t2))))\\
&\qquad\qquad \wedge (\neg \mathrm{eq}(z, y) \Rightarrow \mathrm{eqs}(t2, \mathrm{if}(\mathrm{eq}(z, y)\, t1, \mathrm{if}(\mathrm{eq}(z, x), t1, t2)))))\\
&\equiv (\mathrm{eq}(z, x) \Rightarrow (\neg \mathrm{eq}(z, y) \Rightarrow \mathrm{eqs}(t1, \mathrm{if}(\mathrm{eq}(z, x), t1, t2))))\\
&\quad \wedge (\neg \mathrm{eq}(z, x) \Rightarrow ((\mathrm{eq}(z, y) \Rightarrow ((\mathrm{eq}(z, y) \Rightarrow \mathrm{eqs}(t1, t1))\\
&\qquad\qquad\qquad\qquad \wedge (\neg \mathrm{eq}(z, y) \Rightarrow \mathrm{eqs}(t1, \mathrm{if}(\mathrm{eq}(z, x), t1, t2))))\\
&\qquad\qquad \wedge (\neg \mathrm{eq}(z, y) \Rightarrow ((\mathrm{eq}(z, y) \Rightarrow \mathrm{eqs}(t2, t1))\\
&\qquad\qquad\qquad\qquad \wedge (\neg \mathrm{eq}(z, y)) \Rightarrow \mathrm{eqs}(t1, \mathrm{if}(\mathrm{eq}(z, x), t1, t2))))))\\
&\equiv (\mathrm{eq}(z, x) \Rightarrow (\neg \mathrm{eq}(z, y) \Rightarrow \mathrm{eqs}(t1, \mathrm{if}(\mathrm{eq}(z, x), t1, t2))))\\
&\quad \wedge (\neg \mathrm{eq}(z, x) \Rightarrow (\neg \mathrm{eq}(z, y) \Rightarrow \mathrm{eqs}(t2, \mathrm{if}(\mathrm{eq}(z, x), t1, t2))))\\
&\equiv (\mathrm{eq}(z, x) \Rightarrow (\neg \mathrm{eq}(z, y) \Rightarrow ((\mathrm{eq}(z, x) \Rightarrow \mathrm{eqs}(t1, t1))\\
&\qquad\qquad\qquad\qquad \wedge (\neg \mathrm{eq}(z, x) \Rightarrow \mathrm{eqs}(t1, t2))))\\
&\quad \wedge (\neg \mathrm{eq}(z, x) \Rightarrow (\neg \mathrm{eq}(z, y) \Rightarrow ((\mathrm{eq}(z, x) \Rightarrow \mathrm{eqs}(t2, t1))\\
&\qquad\qquad\qquad\qquad \wedge (\neg \mathrm{eq}(z, x) \Rightarrow \mathrm{eqs}(t2, t2)))))\\
&\equiv \mathrm{true}.
\end{aligned}
$$

From (1), (2), and $\mathrm{eqs}(t3, t4) \equiv \mathrm{true}$ we deduce

$$\mathrm{eqs}(\mathrm{del}(\mathrm{del}(t, x), y), \mathrm{del}(\mathrm{del}(t, y), x)) \equiv \mathrm{true}$$

and thus

$$\mathrm{del}(\mathrm{del}(t, x), y) \equiv \mathrm{del}(\mathrm{del}(t, y), x).$$

Case 3. $t = \mathrm{if}(z, u, v)$. Then

$$
\begin{aligned}
\mathrm{del}(\mathrm{del}(t, x), y) &\equiv \mathrm{del}(\mathrm{if}(z, \mathrm{del}(u, x), \mathrm{del}(v, x)), y)\\
&\equiv \mathrm{if}(z, \mathrm{del}(\mathrm{del}(u, x), y), \mathrm{del}(\mathrm{del}(v, x), y))\\
&\equiv \mathrm{if}(z, \mathrm{del}(\mathrm{del}(u, y), x), \mathrm{del}(\mathrm{del}(v, y), x))\\
&\qquad \text{(by induction hypothesis)}\\
&\equiv \mathrm{del}(\mathrm{if}(z, \mathrm{del}(u, y), \mathrm{del}(v, y)), x)\\
&\equiv \mathrm{del}(\mathrm{del}(t, y), x).
\end{aligned}
$$

Hence by Theorem 2.5, all $\langle$DATA, SET$\rangle$-algebras satisfy (e22). Moreover, each equation $l \equiv r$ satisfied by all consistent and term-generated SET(PX)-algebras belongs to the inductive theory of $\langle$DATA, SET$\rangle$; i.e., $l \equiv r$ is provable in the same way as (e22).

Conclusion

We have developed a completeness theorem for the equational theory of parameterized data types (1.10). Given a parameterized specification PAR, Theorem 1.10 refers to all PAR-algebras and says that their equational theory is just the inductive theory of the target specification equipped with parameter variables. In Section 2 we dealt with consistent algebras where BOOL is interpreted as in propositional logic and the characteristic functions of equality predicates represent identity relations. If a specification SPEC is fork-compatible, then the equational theory of consistent SPEC-algebras agrees with the equational theory of all SPEC-algebras (2.4). The question arises whether this result still holds true for Horn clause logic. The answer is "yes," although additional axioms are needed to transform implications into equations and vice versa (cf. Padawitz, 1987). To go beyond Horn clause logic in data type theory seems to be unreasonable because a richer specification language does not admit initial semantics (cf. Mahr and Makowsky, 1984).

Theorem 2.5 applies Theorem 2.4 to parameterized specifications PAR and can be interpreted as follows: The equational theory of consistent and term-generated target algebras equipped with parameter variables does not exceed the equational theory of PAR-algebras. Maybe, the intersection of both model classes, namely consistent PAR-algebras, has a greater theory. This question is open. Under additional assumptions we obtained a completeness theorem for consistent PAR-algebras (cf. Padawitz, 1984). But that result only applied to *parameter*-sorted equations. So its use is strongly limited and we decided not to include it here.

A last remark concerns fork-compatibility: As our Example 1.2 illustrates, it is a weak requirement. Nevertheless it focuses on the point where logic on the term level (BOOL) extends logic on the clause level (equations or Horn clauses). Fork operations break the classical hierarchy of logic: They map Booleans to non-Booleans and thus allow us to specify non-Booleans under positive as well as negative conditions. It is not the issue of Theorems 2.4 and 2.5 to reduce clausal reasoning to equational reasoning—as, e.g., Paul, 1985, did in his Theorem 2. Instead we aimed at sufficient conditions under which logic on the term level and logic on the clause level can be mixed up in a consistent way. The question whether these conditions are necessary is still open.

Acknowledgment

I thank the referees for careful reading and precise hints at the weak points in the previous version of this paper.

Received February 20, 1985; accepted January 14, 1986

REFERENCES

ARBIB, M. A., AND MANES, E. G. (1979), "Arrows, Structures, and Functors: The Categorical Imperative," Academic Press, New York.

BELL, J. L., AND SLOMSON, A. B., (1971), "Models and Ultraproducts: An Introduction," North-Holland, Amsterdam.

EHRICH, H. D., (1982), On the theory of specifications, implementation and parameterization of abstract data types, *J. Assoc. Comput. Mach.* **29**, 206–227.

EHRIG, H., KREOWSKI, H.-J., THATCHER, J. W., WAGNER, E. G., AND WRIGHT, J. B. (1982), Parameter Passing in Algebraic Specification Languages, *in* "Proc. Workshop in Program Specification," pp. 322–369, Lecture Notes in Computer Science Vol. 134, Springer-Verlag, Berlin/New York.

EHRIG, H., AND MAHR, B. (1985), "Foundations of Algebraic Specification," Vol. 1, "Equations and Initial Semantics," Springer-Verlag, Berlin/New York.

GANZINGER, H. (1983), Parameterized specifications: Parameter passing and implementation with respect to observability, *ACM TOPLAS* **5**, 318–354.

GOGUEN, J. A., THATCHER, J. W., AND WAGNER, E. G. (1978), An initial algebra approach to the specification, correctness and implementation of abstract data types, *in* "Current Trends in Programming Methodology" (R. Veh, Ed., Vol. 4, pp. 80–149, Prentice-Hall, Englewood Cliffs, NJ.

GOGUEN, J. A., AND MESEGUER, J. (1985), Initiality, induction, and computability, *in* "Algebraic Methods in Semantics" (M. Nivat and J. Reynolds, Eds.), pp. 459–540. Cambridge Univ. Press, Cambridge.

KLAEREN, H. A., (1983), "Algebraische Spezifikation: Eine Einführung," Springer-Verlag, Berlin/New York.

LIPECK, U. (1982), "Ein algebraischer Kalkül für einen strukturierten Entwurf von Datenabstraktionen," Dissertation, Universität Dortmund.

MAC LANE, S. (1972), "Kategorien," Springer-Verlag, Berlin/New-York.

MAHR, B., AND MAKOWSKY, J. A. (1984), Characterizing specification languages which admit initial semantics, *Theoret. Comput. Sci.* **31**, 49–59.

NOURANI, C. F. (1981), On induction for programming logic: Syntax, semantics, and inductive closure, *EATCS Bull.* **13**, 51–64.

PADAWITZ, P. (1984), Towards a proof theory of parameterized specifications, *in* "Proc. Int. Symp. on Semantics of Data Types," pp. 375–391, Lecture Notes in Computer Science Vol. 173, Springer-Verlag, Berlin/New York.

PADAWITZ, P. (1985), Parameter preserving data type specifications, *J. Comp. System Sci.* **34**, 179–209.

PADAWITZ, P. (1987), Foundations of Specification and Programming with Horn Clauses, submitted.

PAUL, E. (1985), "On Solving the Equality in Theories Defined by Horn Clauses," Report, Centre National d'Etudes des Télecommunications.

RASIOWA, H., AND SIKORSKI, R. (1968), "The Mathematics of Metamathematics," Polish Scientific Publishers.

RICHTER, M. M. (1978), "Logikkalküle," Teubner, Stuttgart.

THATCHER, J. W., WAGNER, E. G., AND WRIGHT, J. B. (1982), Data type specification: Parameterization and the power of specification techniques, *ACM TOPLAS* **4**, 711–732.